Extreme-scale Electromagnetics for Design and Control of Metamaterials

Presentation by

KELSEY DIPIETRO, TIM WILDEY AND DENIS RIDZAL

Trilinos User-Developer Meeting

October 26, 2022

SANDIA NATIONAL LABS, ALBUQUERQUE, NM

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2022-14898 PE

Purpose and outline

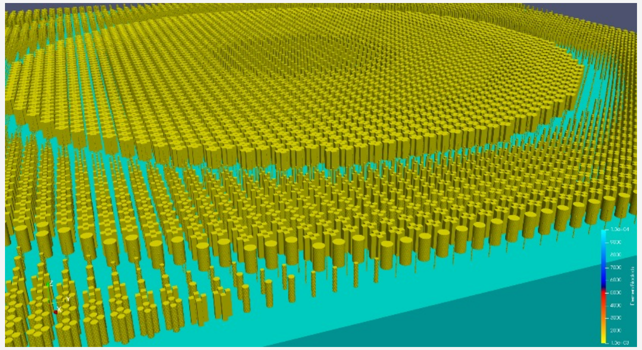
Highlight the interactions between **Trilinos** and **Mirage**, a research project and an easy-to-use high-performance software for the design of electromagnetic metamaterials.

HIGHLIGHTS

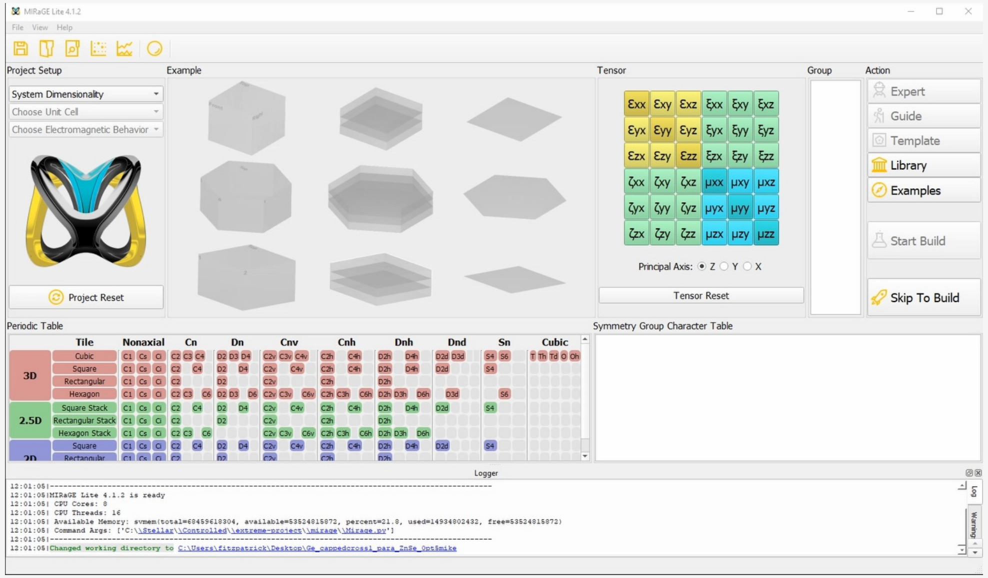
- **Mirage**: Team, goals and software capabilities.
- **Trilinos**-based components: **$(\mathbb{FEM})^3$** and **MrHyDE**.
- A sampling of **interactions** and **collaborations** between Trilinos and Mirage.
- New products: Extreme-scale meshing with **Zellij**, plug-n-play Trilinos with **containers**.
- Looking into the future: Reducing **memory** footprint.

Mirage is a DARPA-funded research project and software

GOAL Electromagnetic devices featuring nanoscale structures.



EXAMPLE A micron-thin lens composed of millions of *atoms*.
PURPOSE Weight/space savings.



METHOD Design *atoms*, tile, simulate, optimize.

TEAM

Edgar Bustamante, Cesar Valle, Kelsey DiPietro
Denis Ridzal, Tim Wildey & Ihab El-Kady (PI)
Sandia National Labs, Albuquerque, NM, USA

David Fitzpatrick, Ryan Chilton, Tony Wilson,
Mehmet Su & Jane Burward-Hoy
Stellar Science Ltd. Co., Albuquerque, NM, USA

WITH MANY CONTRIBUTORS

MIRaGE

Multiscale Inverse Rapid Group-theory for Engineered-metamaterials

Sandia National Laboratories

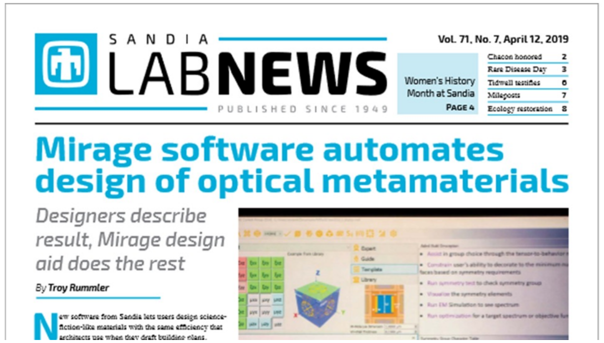
mirage-software.com

stellar science

Exceptional service in the national interest

Stellar Scientific Software Solutions

Mirage is a product of interactions with the world’s best


SANDIA LABNEWS
PUBLISHED SINCE 1949
Vol. 71, No. 7, April 12, 2019
Women's History Month at Sandia PAGE 4

Mirage software automates design of optical metamaterials

Designers describe result, Mirage design aid does the rest

By Troy Rummler

AWARDS AND RECOGNITIONS

2018: DARPA D60 Showcase
2019: R&D 100 Award
2020: Laser World Focus Featured Article
2020: Laser World Focus Platinum Award
2020/2021: FLC Regional Tech Development Nomination
2021: HPC-Report Featured Article
2021: FLC Tech Transfer Mid Continent Award
2021: SNL Partnerships Annual Report Featured Article
2022: FLC National Award Nomination

Mirage large-scale simulation is powered by

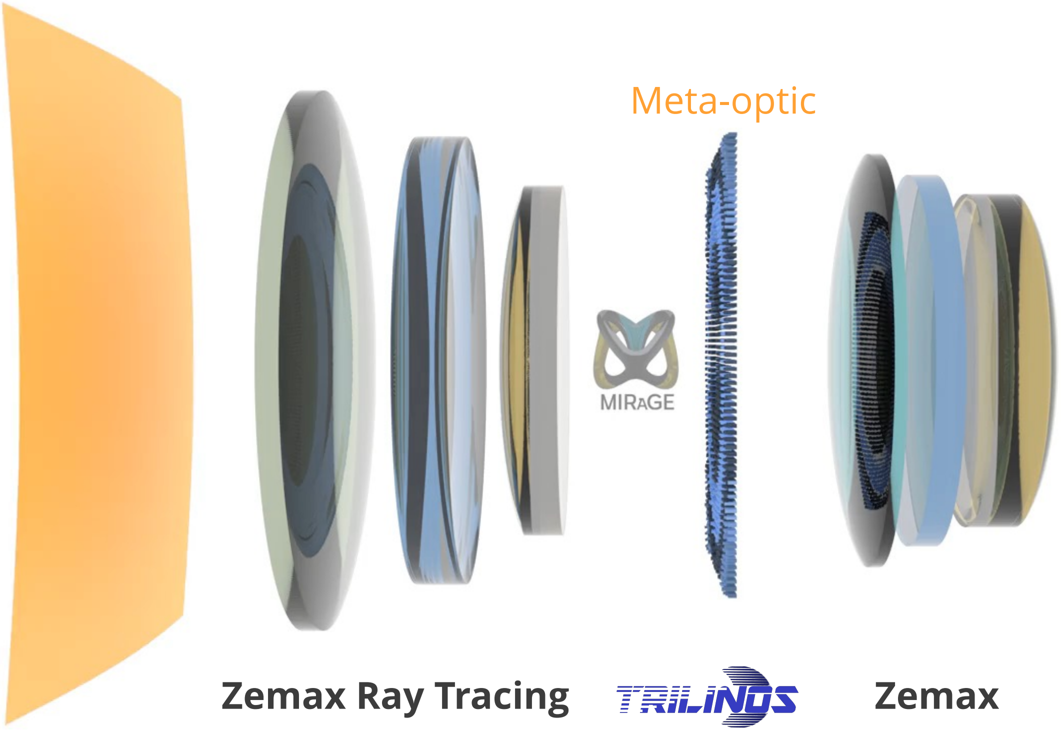
TRILINOS

Mirage capabilities currently support night-vision R&D

CAPABILITIES

Geometry/CAD
Atom meshing with Cubit
Frequency-domain EM simulation
Group-symmetry design
Nonlinear/dispersion modeling
Device layout and tiling
Extreme-scale meshing via stitching
Extreme-scale time-domain EM simulation
Shape and topology optimization
Design and control of EM sources
Lens design with interface to Zemax
Laptop/Workstation/HPC

NIGHT-VISION OPTICAL TRAIN



TRILINOS SUPPORTS EXTREME-SCALE EM COMPONENTS

kokkos

MueLu Multigrid Framework

ROL RAPID OPTIMIZATION LIBRARY

Panzer

Seacas

etc.

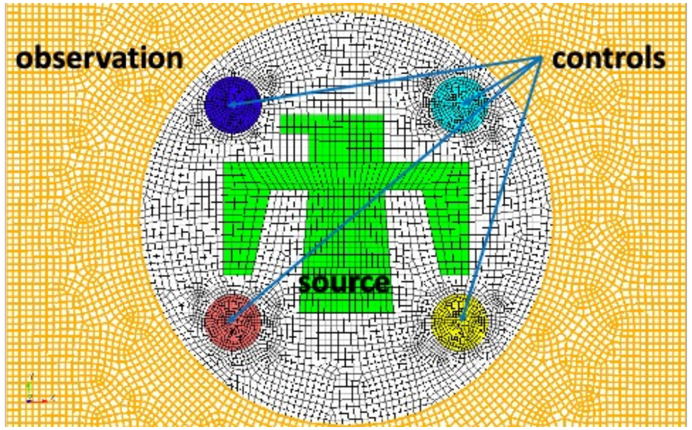
(FEM)^3

MrHyDE

Small-scale lens simulation

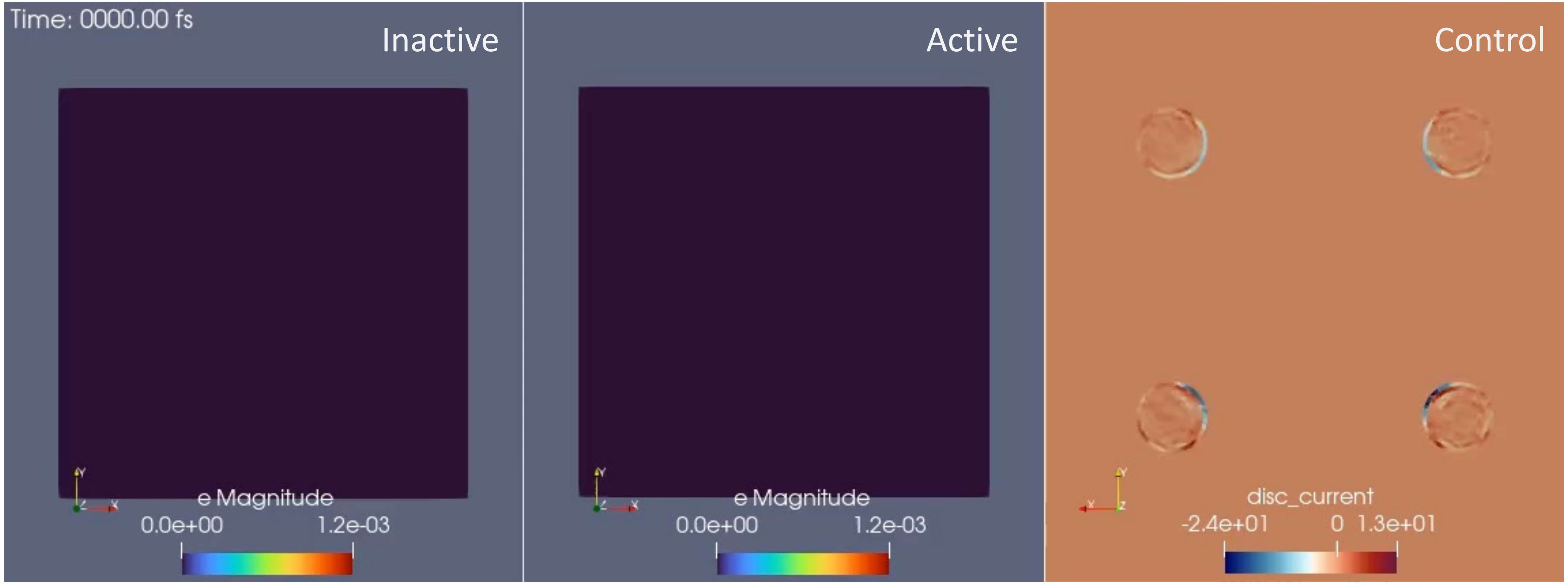
- Cross section of a lens.
- Planewave source illuminating the lens from bottom of domain.
- Roughly 1 billion variables.
- Runtime: 2 hours on 50 HPC nodes.
- Primary focal spot, secondary focal spot, edge effects.
- Real optics need simulations with **trillions of variables**.

Electromagnetic source design and control



GOAL Hide the thunderbird-shaped EM source by applying a current source in the four circular control regions.

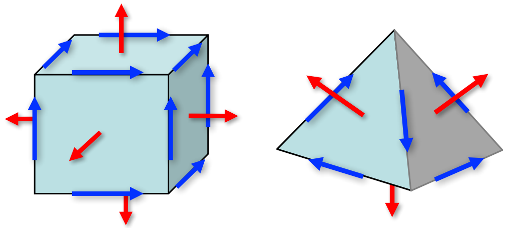
METHOD Solve an optimal control problem.



$(\mathbb{FEM})^3$ is Mirage's 1st-gen extreme-scale engine

Maxwell's Equations

$$\frac{\partial \mathbf{D}}{\partial t} - \nabla \times \mathbf{H} = -\mathbf{J},$$
$$\nabla \cdot \mathbf{D} = \rho,$$
$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = \mathbf{0},$$
$$\nabla \cdot \mathbf{B} = 0.$$



Weak Form

$$\left(\frac{\partial \mathbf{B}}{\partial t}, \mathbf{C}\right)_\Omega + (\nabla \times \mathbf{E}, \mathbf{C})_\Omega = 0,$$
$$\frac{1}{c^2}\left(\frac{\partial \mathbf{E}}{\partial t}, \mathbf{F}\right)_\Omega - (\mathbf{B}, \nabla \times \mathbf{F})_\Omega = (\mathbf{n} \times \mathbf{B}, \mathbf{F})_{\partial\Omega} - \mu (\mathbf{J}, \mathbf{F})_\Omega .$$

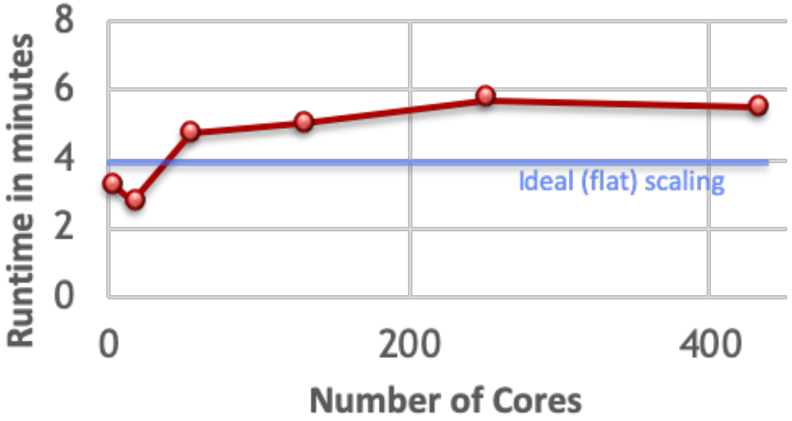
Mixed algebraic system

$$Q_B \frac{\partial \mathbf{B}_h}{\partial t} + K\mathbf{E}_h = \mathbf{0},$$
$$\frac{1}{c^2} Q_E \frac{\partial \mathbf{E}_h}{\partial t} - K^t \mathbf{B}_h = -\mu \mathbf{J}_h$$

We solve the **mixed system** using GMRES preconditioned with multigrid for the E-field Schur complements.

Super cell	Number of Equations	Number of cores
1x1x1	468,238	2
2x2x2	2,844,939	16
3x3x3	9,451,197	54
4x4x4	22,289,564	128
5x5x5	43,202,452	250
6x6x6	79,414,389	432

Total runtime, fixed work per core



- Discretize E-field using **Nedelec edge elements**, and B-field using **Raviart-Thomas face elements**.
- Discretize time using **implicit methods**.
- Use variants of the scalable MueLu **RefMaxwell** multigrid preconditioner to solve edge Schur complement systems.

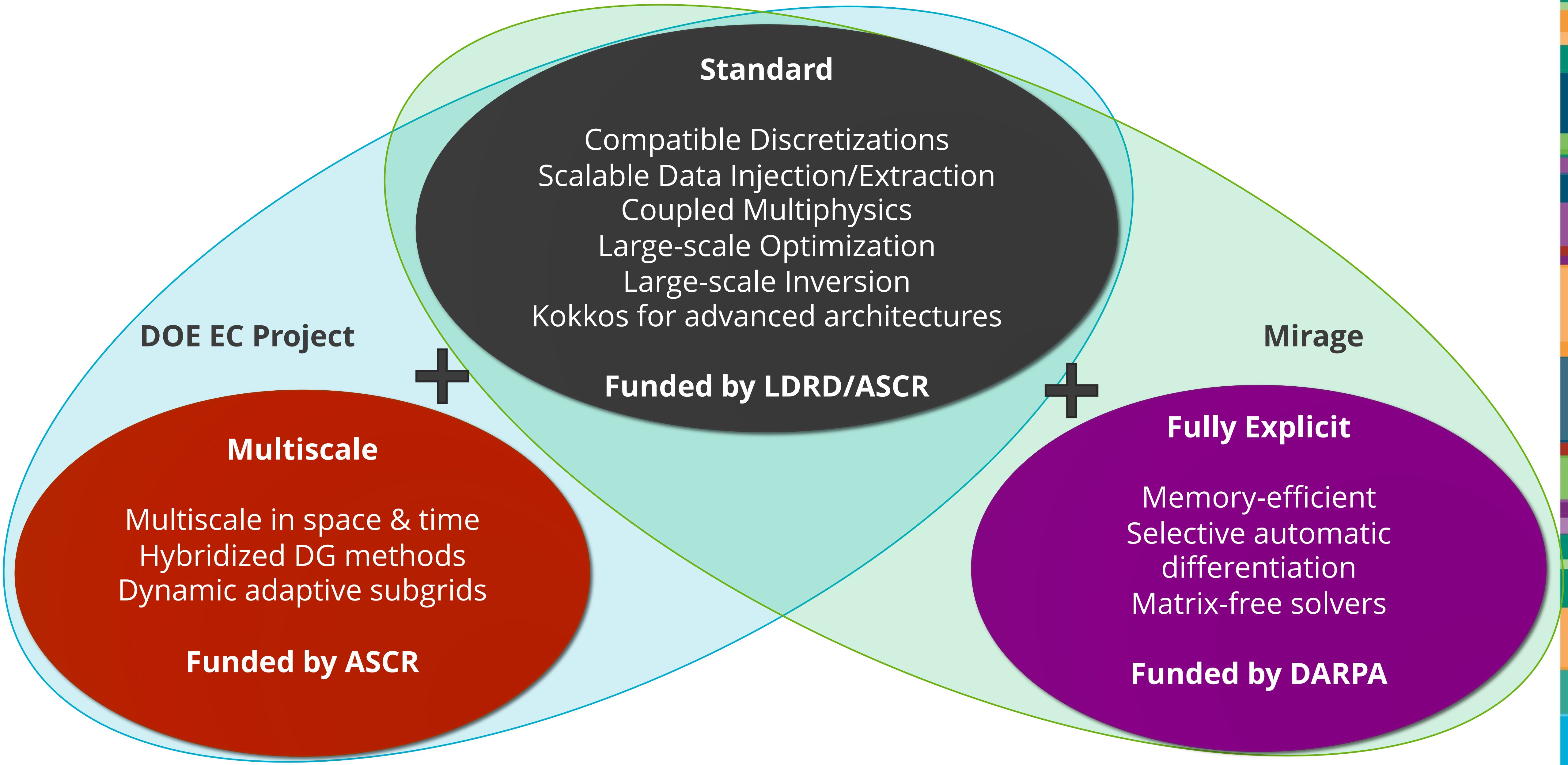
HISTORY

$(\mathbb{FEM})^3$ grew out of **miniEM**, and it was a great launch platform for the Mirage time-domain capability. Moving toward **trillions of finite elements** requires a more fine-grained tool.

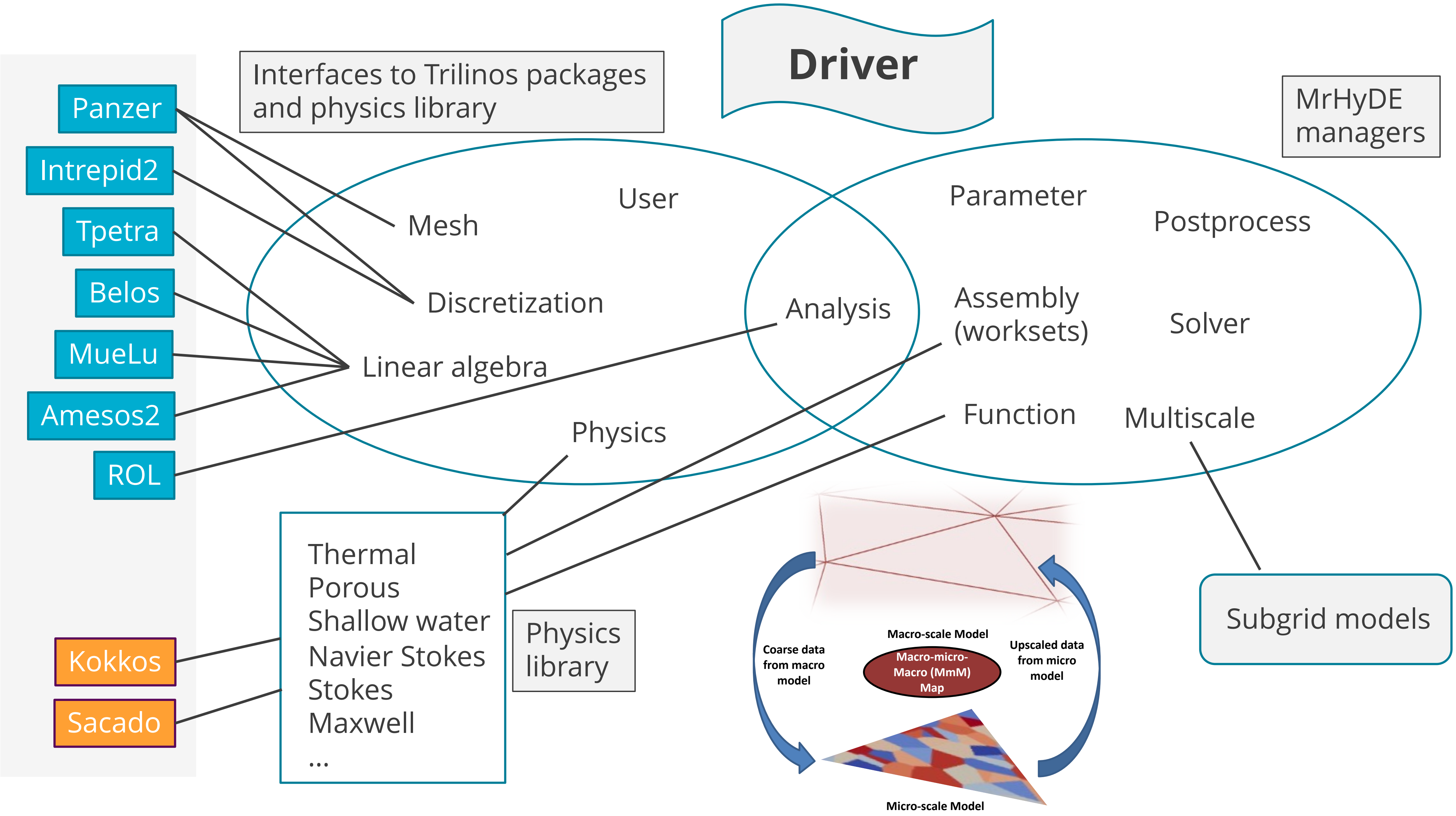
MrHyDE provides Mirage's 2nd-gen extreme-scale engine

- A C++ framework designed and optimized for solving Multi-resolution Hybridized Differential Equations (MrHyDE).
- Provides an interface to powerful Trilinos tools within a user-friendly framework.
- Portability with performance from laptops, to MPI-based clusters, to heterogeneous nodes, to MPI+X.
- Ability to extract and inject data to enable data-informed physics-based simulations.
- A modular and flexible environment for solving transient nonlinear multiphysics and multiscale systems.
- Extensive set of examples/regression tests to maintain software quality and guide new users.

MrHyDE has three operating modes



MrHyDE stands on the shoulders of Trilinos



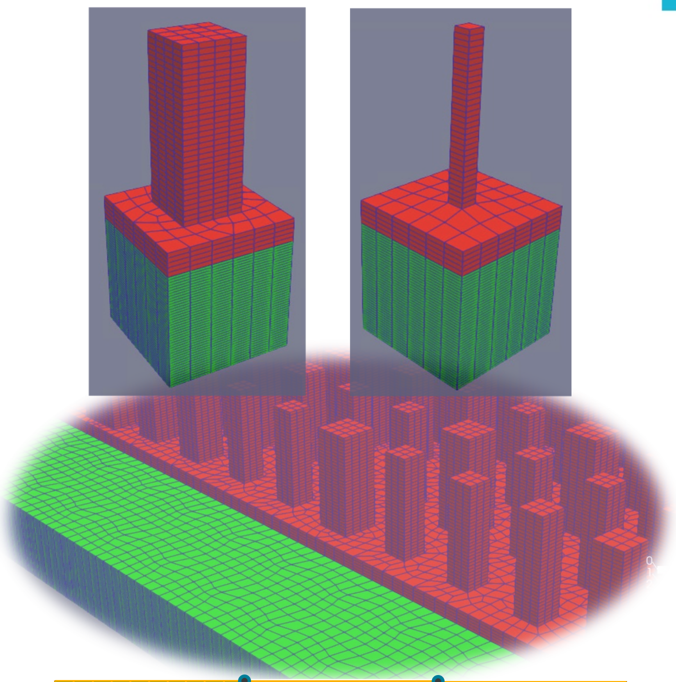
Mirage has given back to Trilinos

- **UNIQUE FEATURE?** Mirage's smallest "useful" problem requires 1 billion finite elements.
- Uncovered numerous tough bugs and missing features.
  - Provided well-documented bug reproducers and/or bug fixes.
  - Written unit tests.
  - In some cases, funded Trilinos developers to fix bugs and develop unit tests.
  - A few examples:
    - Panzer: New parallel tiebreak for correct multiphysics DOF indexing. (with Eric Cyr)
    - Tpetra: Hash table error in createOnetoOne map. (with Karen Devine and Mark Hoemmen)
    - Panzer: 32-bit integer limit in DOF Manager. (with Roger Pawlowski)
    - MueLu: Various performance improvements in RefMaxwell. (with Christian Glusa)
    - Panzer: Quadratic runtime scaling in DOF matching for periodic boundary conditions. (with Bryan Reuter, Roger Pawlowski and Eric Cyr)
    - STK: Quadratic memory scaling in parallel mesh database construction. (with Alan Williams)
- Funded the development of new capabilities, such as **Zellij** and **Trilinos Containers**.
- Motivated important new research, such as in I/O throughput and **memory footprint**.

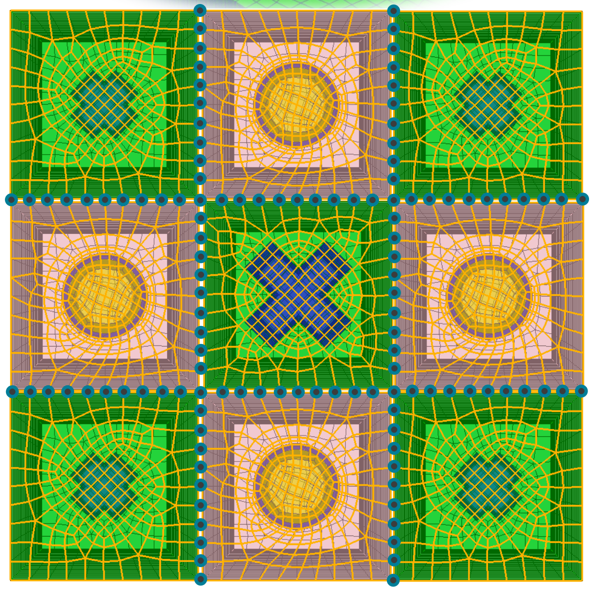
Seacas tool Zellij

MOTIVATION For lenses, Mirage uses hexahedral meshes with billions of elements. There were no existing solutions suitable for the required extreme-scale mesh generation.

SOLUTION Exploit the **unit-cell structure** of the device layout and **stitch together** the conformal mesh from a dictionary of individual unit-cell meshes.



- Greg Sjaardema came to the rescue. Greg developed a new **Seacas** tool, called **Zellij**, for extreme-scale mesh stitching.
- Zellij is a **mesh concatenation** application for generating a mesh consisting of a **lattice** containing one or more unit-cell template meshes.
- The lattice is a **two-dimensional arrangement** of the unit-cell template meshes.
- *Note*: We also worked with Steve Owen to develop an *unstructured mesh stitching* solution as part of the Sculpt application.



Seacas tool Zellij

- Given a dictionary of files and a matrix (mapping) of numbers or symbols to the files, Zellij concatenates the files **without building the entire mesh in memory**.

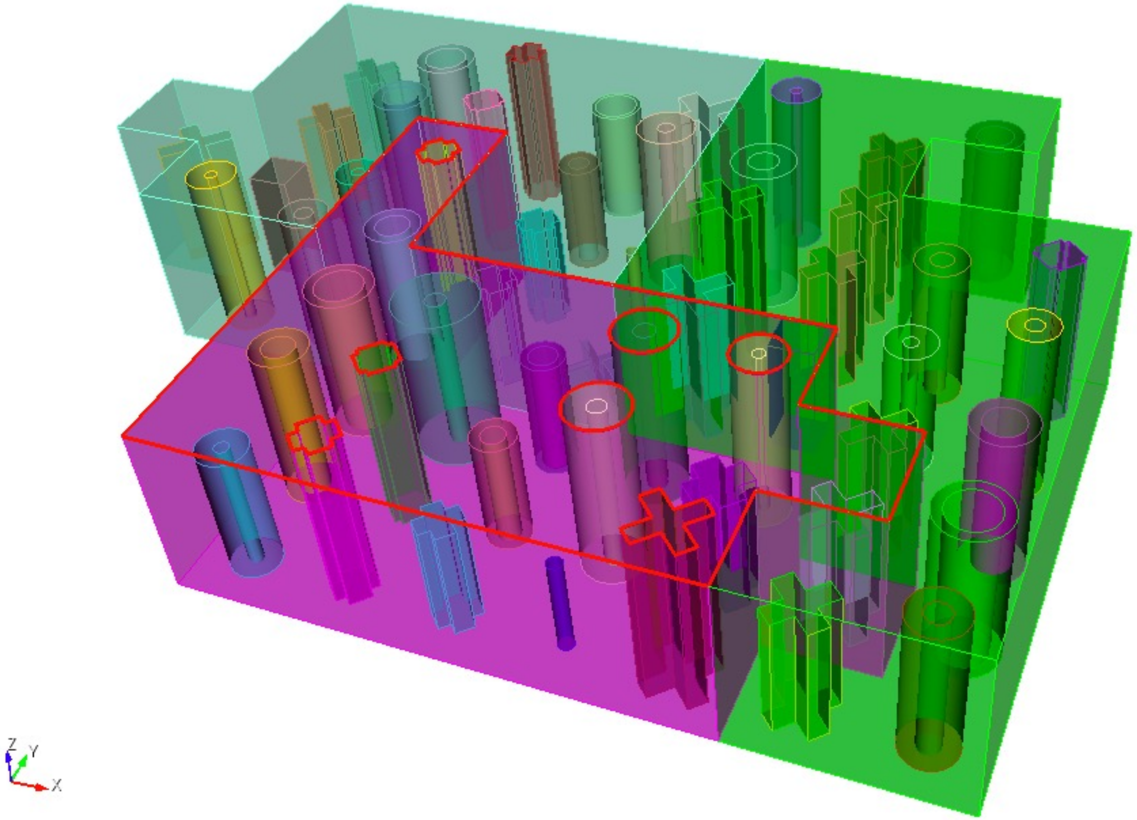
```
BEGIN_DICTIONARY
- "../zellij-example/xatom-1b.e"
| "../zellij-example/xatom-Y.e"
+ "../zellij-example/xatom-X.e"
* "../zellij-example/xatom-2b.e"
END_DICTIONARY
```

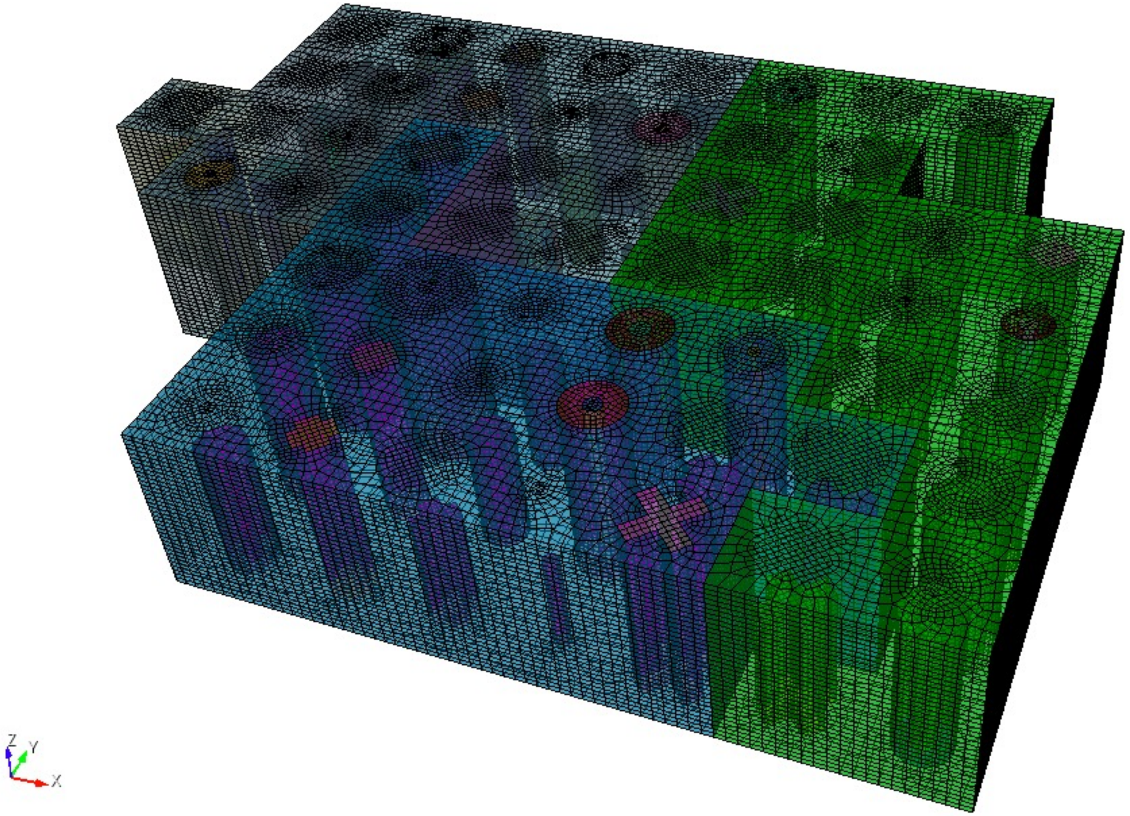
+

```
BEGIN_LATTICE 5 5 1
- | + | -
| + + + |
+ + * + +
| + + + |
- | + | -
END_LATTICE
```

- The number of output files is **specified by the user** … it can be 1 (generates single file), 2, etc., up to the number of inputs in the lattice, so 25 in the example above.
- Can be used in serial (if resources are limited) or in parallel. In either case, there are no memory limitations, and Zellij will produce a **mesh of virtually any size** given sufficient runtime.
- Zellij can perform a **coarse-grained load balancing** of the output meshes, by applying Hilbert Space Filling Curves (HSFC) or other algorithms **to the unit-cell lattice**.

Zellij: Coarse-grained load balancing





- Nearly **optimal load balancing** for unit-cell based meshes.
- Mesh generation through stitching is **super-fast and runtime & memory scalable**.

SEMS Trilinos Container for Mirage

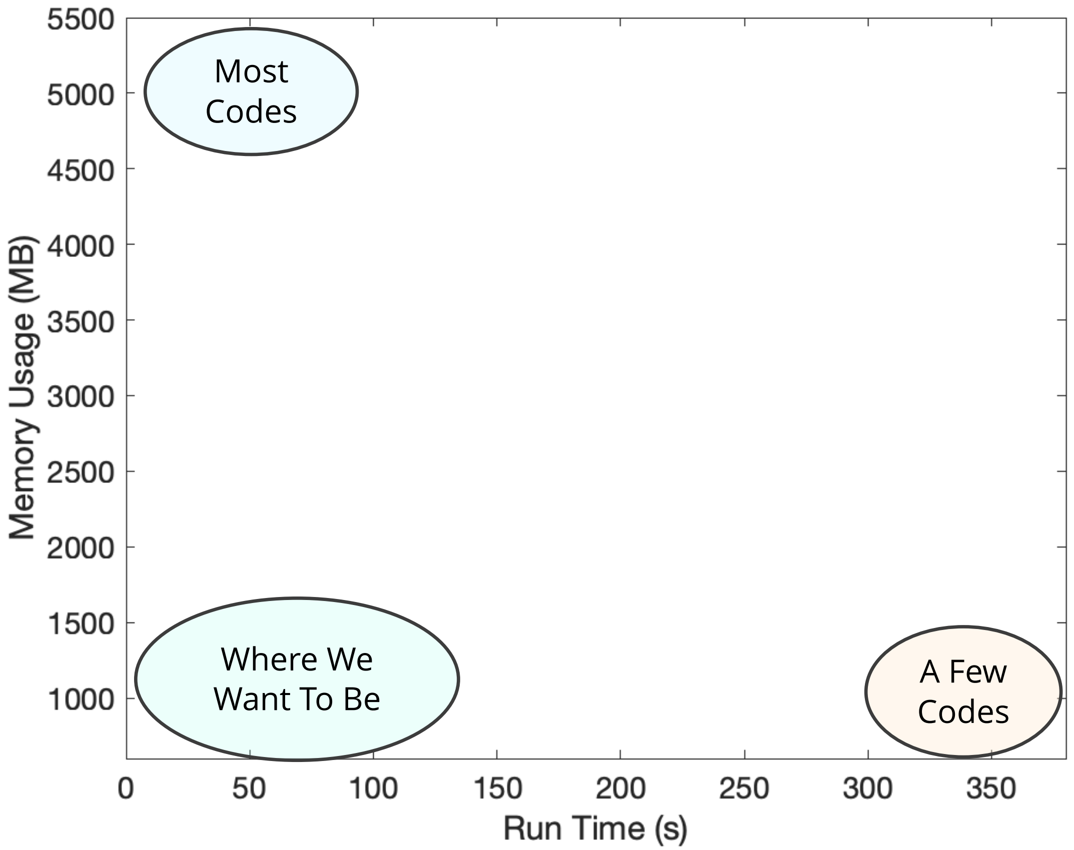
- Deploying HPC solutions on a **variety of platforms** is difficult:
  - Compilers, third-party libraries (TPLs), parallel execution (MPI).
- Containers encapsulate the **full runtime environment (operating system)** and the **parallel application executable**.
- More **agile, scalable and portable** than virtual machines.
- **Sandia's Software Engineering (SEMS)** team has helped us develop container solutions for:
  - **Docker** – the most commonly used container tool.
  - **Singularity** – enables use of Docker images on HPC platforms.
- Status:
  - Completed **Docker** container based on CentOS 8 and optimized TPLs for Mirage.
  - Tested on **Mac** and **Windows**: scalable MPI parallel execution.
  - OCI-compatible: Can be run with Podman, Singularity, etc.
  - **Singularity** HPC workflow is almost ready, requires testing.

Work with Elliott Ridgway and many others from SEMS.

Looking forward: Reducing memory footprint

R-adaptivity to enable compression of elementary computations in extreme-scale finite element simulators

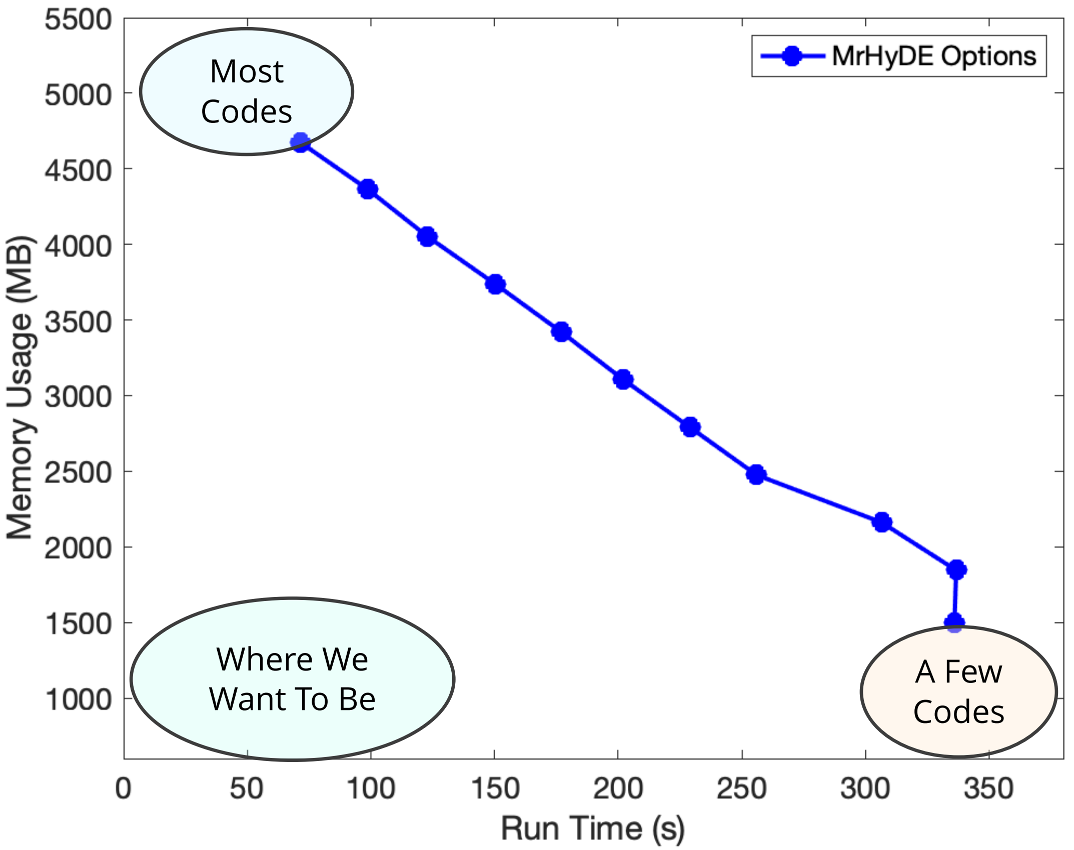
- Most applications/codes have been willing to sacrifice memory for performance.
    - Limits the size of the problems we can run on Sandia resources.
- A few, particularly those targeting GPU platforms, will sacrifice performance for memory.
- ***We aim to challenge the notion that we can't have both.***



Looking forward: Reducing memory footprint

R-adaptivity to enable compression of elementary computations in extreme-scale finite element simulators

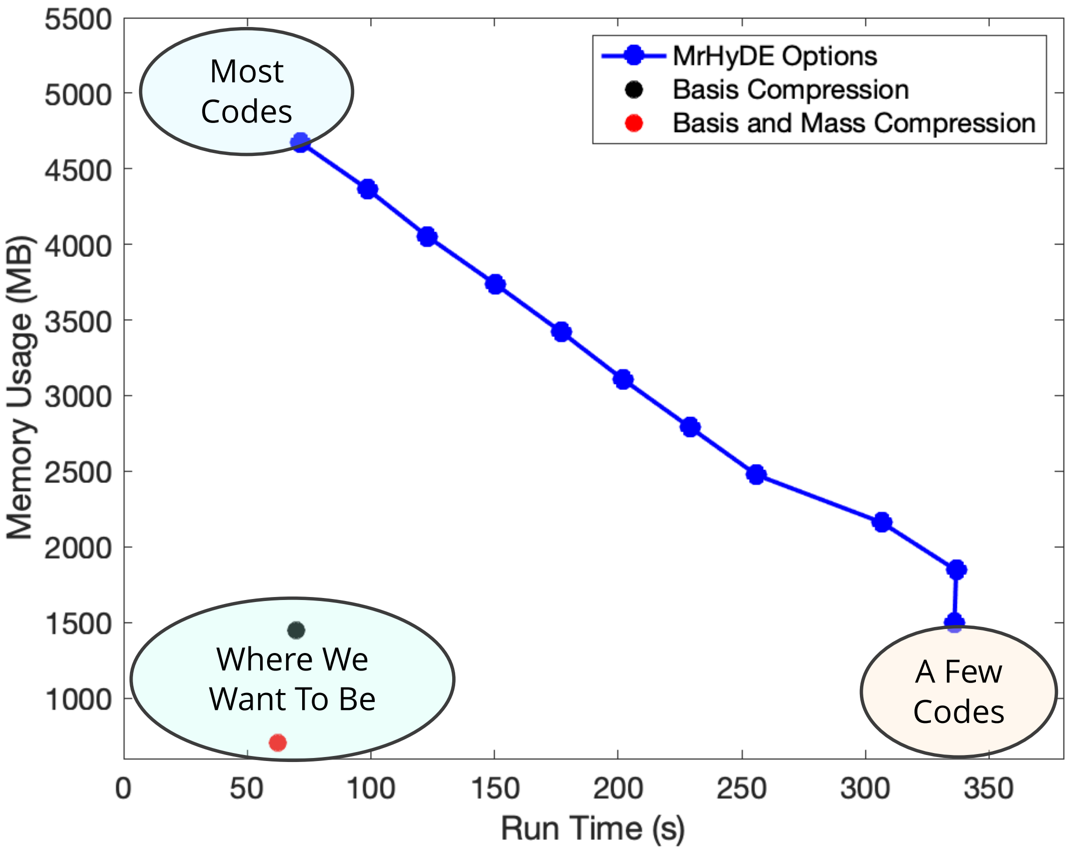
- Most applications/codes have been willing to sacrifice memory for performance.
  - Limits the size of the problems we can run on Sandia resources.
- A few, particularly those targeting GPU platforms, will sacrifice performance for memory.
- ***We aim to challenge the notion that we can't have both.***
- We can change the ratio of properties being stored to recomputed (blue line).



Looking forward: Reducing memory footprint

R-adaptivity to enable compression of elementary computations in extreme-scale finite element simulators

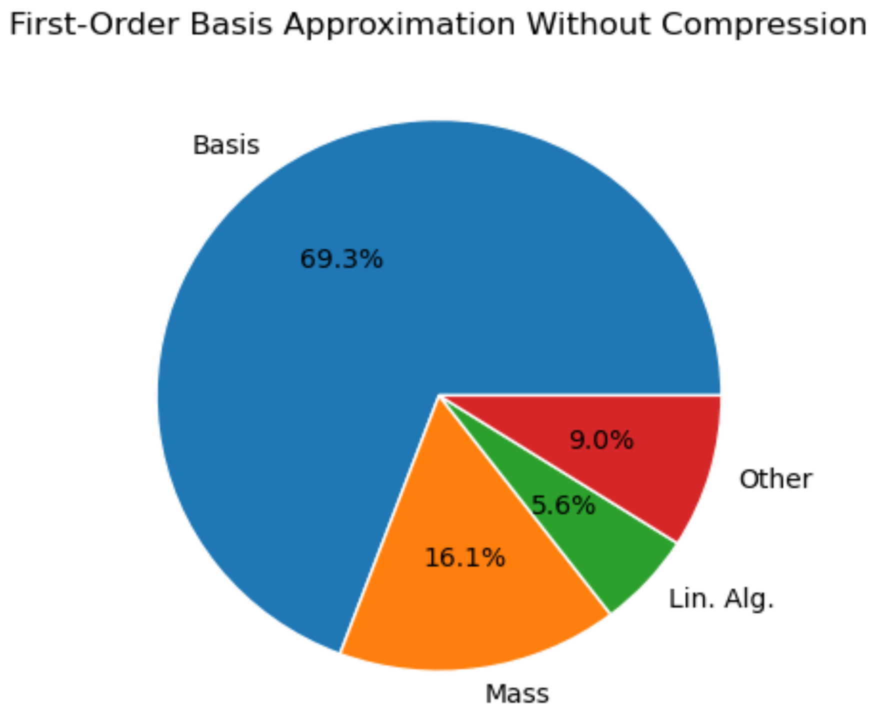
- Most applications/codes have been willing to sacrifice memory for performance.
  - Limits the size of the problems we can run on Sandia resources.
- A few, particularly those targeting GPU platforms, will sacrifice performance for memory.
- ***We aim to challenge the notion that we can't have both.***
- We can change the ratio of properties being stored to recomputed (blue line).
- Pose as a data science challenge: achieve up to 98% data-compression of finite element computations on realistic meshes (red and black data in figure).
- This allows us to run problems ***many times larger*** than previously possible ***without sacrificing runtime or accuracy***.
- Compress the number of unique finite element quantities to be stored for a given problem.

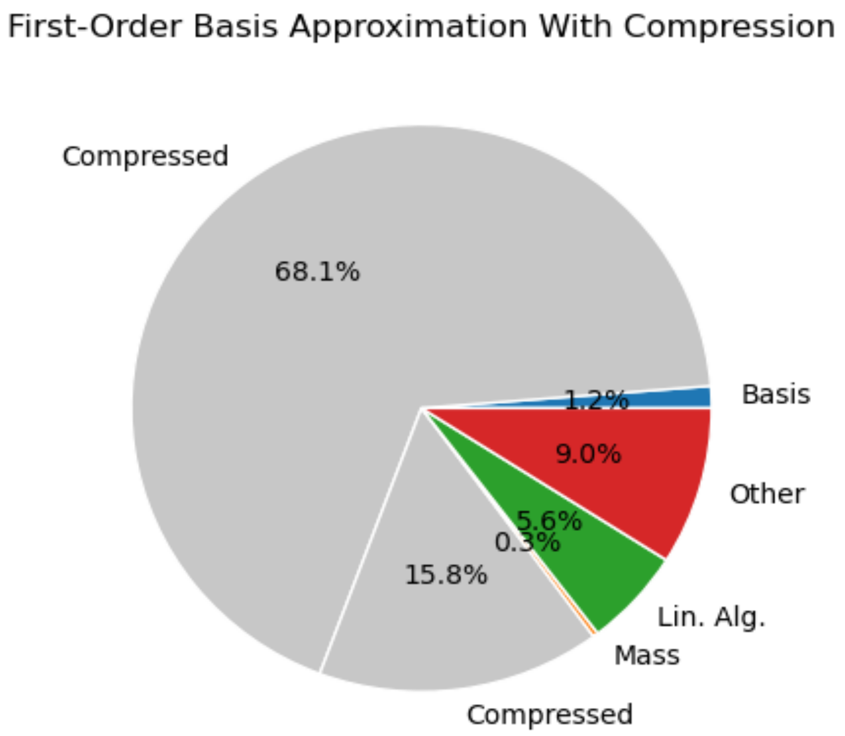


Looking forward: Reducing memory footprint

Memory usage for the components of a finite element simulation.

- Run on a mesh of 540k elements on a single core.





Compression in the two largest contributors causes a 84% reduction in overall memory usage.
Even more prominent for higher-order elements (92%).

Looking forward: Reducing memory footprint

R-adaptive Techniques	Main Idea	Applications
Moving mesh partial differential equations (MMPDEs) [1,2,3]	Mesh points are determined as the solution to a gradient flow equation of a meshing functional. • Control comes from the metric tensor and meshing functional. • Gives a parameterization of the mesh based on minimizing over specific quantities.	• Adaptivity on surfaces and 2D domains. • Mostly in response to PDE behavior such as shocks and singularities.
Sandia's Mesquite Code [4]	Improves mesh quality for unstructured meshes using the target optimization paradigm. • Focused on size and shape of mesh elements.	• Fully focused mesh quantity to avoid issues when running PDE solutions.

Use ideas from moving mesh adaptivity to create meshes with increased redundancy of low-level finite element data!

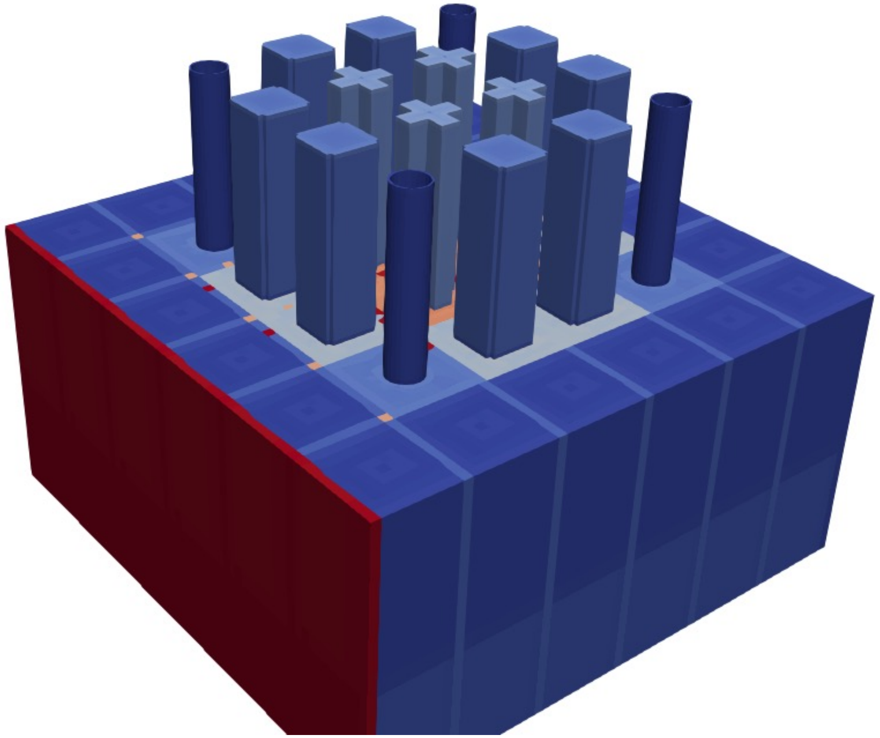
[1] **K. L. DiPietro**, R. D. Haynes, W. Huang, A. E. Lindsay, and Y. Yu, "Moving mesh simulation of contact sets in two dimensional models of elastic–electrostatic deflection problems," Journal of Computational Physics, vol. 375, pp. 763–782, 2018.
[2] W. Huang, L. Kamenski, and H. Si, "Mesh smoothing: An mmpde approach," 2015.
[3] A. Kolasinski and W. Huang, "A surface moving mesh method based on equidistribution and alignment," Journal of Computational Physics, vol. 403, p. 109097, 2020.
[4] L. Freitag, T. Leurent, P. Knupp, and D. Melander, "Mesquite design: issues in the development of a mesh quality improvement toolkit." 3 2002.

Looking forward: Reducing memory footprint

EXAMPLE

- Create a database for elements in a semi-structured mesh.
- For example, the meshes for Mirage are unstructured in the xy-plane, but extruded in z.
- This yields tremendous redundancy in the basis and mass matrix information between the elements.
- By compressing this information, we reduce memory by up to 98% and reduce runtime by 10%.
- Compression rates are even more dramatic for higher-order discretizations.
- However, not all meshes have this much redundancy.
- ***Given a mesh with less redundancy, can we modify the mesh to maximize redundancy while maintaining accuracy?***

Goal: Improve the redundancy in unstructured meshes to develop **finite element methods with the speed and memory footprint of finite differences**.



The colors indicate the amount of unique Jacobian information for an extruded mesh. There are approximately 3,330 unique color IDs for a mesh of 750k elements.